Wireless Sensor Network using Particle Swarm Optimization

Shekh Md Mahmudul Islam[1], Mohammad Anisur Rahman Reza and Md Adnan Kiber

Department of Applied Physics, Electronics & Communication Engineering,

University of Dhaka, Dhaka, Banngladesh-1000

Email: mahmud_apececadet@yahoo.com

***Abstract*— Wireless sensor network (WSN) is becoming progressively important and challenging research area. A Wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical and environmental conditions and to co-operatively pass their data through the network to a main location. Wireless sensor consists of small low cost sensor nodes, having a limited transmission range and their processing, storage capabilities and energy resources are limited. The main task of such a network is to gather information from a node and transmit it to a base station for further processing.WSN has different issues such as optimal sensor deployment, node localization, base station placement, location of target nodes, energy aware clustering and data aggregation. Recently researchers around the world are applying bio-inspired optimization algorithm known as particle swarm optimization (PSO) for increasing efficiency in the WSN issues. This paper describes the use of PSO algorithm for optimal sensor deployment in WSN.**

***Index Terms*— Wireless sensor network (WSN), Particle Swarm Optimization (PSO), base station placement**

I. Introduction

Wireless Sensor Network (WSN) are an emerging technology[1] that has potential application in surveillance, environment and habitat monitoring ,structural monitoring and healthcare, and disaster management[2]. A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to *monitor* physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling *control* of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; The WSN is built of “nodes” – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning “motes” of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hoop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding. Today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.WSN consists of four main components: a radio, a processor, sensor and battery. In most deployments sensor nodes have self-organizing capabilities to form an appropriate structure. A WSN monitors an environment by sensing its physical properties .It is a network of tiny, inexpensive autonomous nodes that can acquire process and transmit sensory data over wireless medium. One or more powerful base station serves as the final destination of the data. The properties of WSNs that pose technical challenge include dense ad-hoc deployment, dynamic topology. WSN issues such as node deployment, localization, energy-aware clustering and data aggregation are often formulated as optimization problems. Traditional analytical optimization technique requires enormous computational effort, which grows exponentially as the problem size increases. An optimization method that requires moderate memory and computational resources and yet produces good results is desirable, especially for implementation on an individual sensor node. Bio-inspired optimization methods are computationally efficient alternatives to analytical methods. Particle swarm optimization is a popular multi-dimensional optimization technique. Ease of implementation, high quality of solutions, computational efficiency and speed of convergence are strength of PSO. This paper focuses on sensor deployment using Particle Swarm Optimization (PSO) algorithm. One of the fundamental issues that arise in WSN is coverage area in addition to location identification, tracking and deployment. In this coverage the nodes have the effective responsibility to cover the predefined area. The most effective approach for sensor deployment is to place sensor in such a manner that the maximal network coverage is achieved.

II. Particle Swarm Optimization

PSO is a population based optimization technique developed by Eberhart and Kennedy in 1995[3]. In PSO, the potential solutions, called particles, fly through the problem space following the current optimum particles. Each particle keeps track of its coordinates in the problem space, which

DOI: 03.LSCS.2013.1.502

are associated with the best solution (fitness) it has achieved so far (the fitness value is also stored). This value is called *pbest*. Another *best* value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the swarm called *gbest*. Each individual in PSO flies in the search space with a velocity which is dynamically adjusted according to its own flying experience (*pbest*) and its companion's experience (*gbest*), so that the individuals of the population can be expected to move towards better solution areas. Each individual is treated as a volume-less particle in the D-dimensional search space. The particles are manipulated according to the following equations [4]

$$V_{id} = w*V_{id} + c_1*rand(\ )*(p_{id} - X_{id}) + c_2*Rand(\ )*(p_{gd} - X_{id}) \qquad (1)$$

$$X_{id} = X_{id} + V_{id} \qquad (2)$$

where c_1 and c_2 are positive constants (learning factors), and rand() and Rand() are two random functions in the range [0,1]; $X_i = (X_{i1}, X_{i2}, ..., X_{iD})$ represents the i^{th} particle; $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$ represents the best previous position (the position giving the best fitness value) of the particle; the symbol *g* represents the index of the best particle among all the particles in the population; $V_i = (V_{i1}, V_{i2}, ..., V_{iD})$ represents the rate of the position change (velocity) for particle *i*. Equations (1 and 2) are the equation describing the flying trajectory of a population of particles. Equation (1) describes how the velocity is dynamically updated and Equation (2) the position update of the "flying" particles. Equation (1) consists of three parts. The first part is the momentum part. The velocity can't be changed abruptly. It is changed from the current velocity. The second part is the "cognitive" part which represents private thinking of itself - learning from its own flying experience. The third part is the "social" part which represents the collaboration among particles - learning from group flying experience. The commonly used PSOs are either global version or local version. In the global version of PSO, each particle flies through the search space with a velocity that is dynamically adjusted according to the particle's personal best performance achieved so far and the best performance achieved so far by all the particles. In the local version of PSO, each particle's velocity is adjusted according to its personal best and the best performance achieved so far within its neighborhood. The neighborhood of each particle is generally defined as topologically nearest particles to the particle at each side. A local version of PSO with time varying inertia coefficient is used in this work is shown in Fig 1.

II. Optimal Wsn Deployment

WSN problem refers to determining position for sensor nodes such that the desired coverage, connectivity and energy efficiency can be achieved with as few nodes as

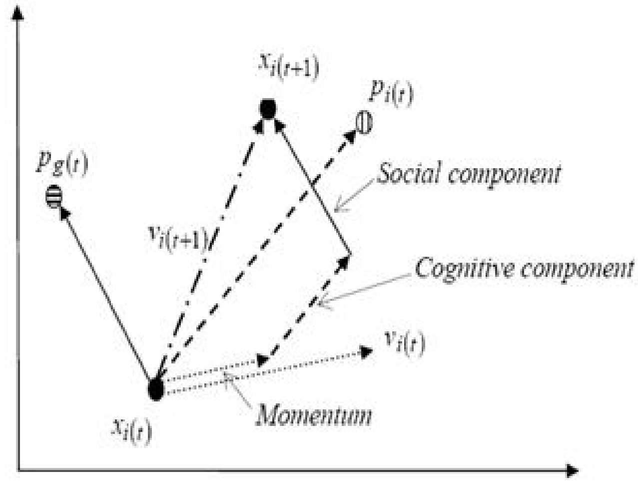


Fig 1: Geometric illustration of particle movement in PSO

possible[5].Events in an area devoid of an adequate number of sensor nodes remain unnoticed; and the areas having dense sensor population suffer from congestion and delays. Optimally deployed WSN assures adequate quality of service, long network life and financial economy. Available PSO solutions to the deployment problem are computed centrally on a base station for determining position of sensors.

A. Sensor Coverage

A sensor placed on a location point(X1, Y1) can cover a location point (X2, Y2), if the Euclidian distance between the two points is:

(X1-X2)2-(Y1-Y2)2<=r2

Where, r is the sensing range of the sensor. The mean value of the location point (Xi,Yi) for i=1,2,….M is represented by (mx,my).Sensor node is the centered of location points it has to cover. The distance between the sensor node and the farthest location point denote the sensing range r.Area, A is divided into R regions and each region is placed with sensor nodes by minimizing Euclidian distance between location points and their closest centroid. Area A is covered with R sensor nodes. The coverage problem can be formulated as an optimization problem and defined as: P is the set points and R is the fixed no of sensors, the optimum location for deploying all R sensors such that every location point is covered.

III. Problem Formulation

The main objective of the present work paper is to minimize the distance between the neighboring nodes, maximizing coverage in the network, while simultaneously satisfying all constraints.

1. All sensor nodes are homogenous and have mobility
2. We assume that deployed sensor nodes can fully cover the sensing fields Sensing coverage and communication coverage of each node assumed to have circular shape without any irregularity.
3. The design variables are two dimensional co-ordinates of the sensor nodes
4. All the sensor nodes cover equal sensing field areas

DOI: 03.LSCS.2013.1.502

The above are common assumption for many sensor network applications.

A. Flow chart

The Flow chart contains a recursive iteration loop and can be described by the following pseudocode.Fitness,F; F depends on the Euclidian distance between the sensor node and the nearest centroid. Calculate fitness for each particle. Among the swarm, the particle with least fitness is considered as the global best particle as it is closest to the optimum solutions. The swarm is said to have accomplished the task if all the particles in it have acquired fitness less than or equal to the range of sensors incorporated in the network. In particle swarm optimization we perform the following actions [Fig. 2]

1. Network information and algorithm parameters—inertia, weight, learning factor, velocity boundary value and the largest iterative numbers are initialized. Array of particles are initialized with random position and velocity vector.
2. Find the distance of the interest point to its nearest sensor. Fitness is evaluated for every particle at its current position using Euclidian distances.
3. Minimize the fitness value; ideally the fitness value should be equal to zero, where the distance between the interests points with their nearest sensors are within the sensor's sensing range. If the fitness of the particle is lesser than that of the best particle, then the particle would be the best particle for the next move, and fitness of that particle is taken as best fitness.
4. Each particle is made to modify its current position and current velocity.
5. If the next position of the particle is the best, then the particle chooses a new position, otherwise the same algorithm is continued.
6. The process is repeated in iteration, until all the particles communicate with each other and generate maximum coverage.

B. PSO parameters

For the proposed method the number of particles are taken as 40 and the learning factor $C_1=C_2=2$.An inertia weight factor is linearly reduced as the search proceeds from 0.9 to 0.4[Xiahui et al(2004)].The maximum velocity and maximum iterations[Bo Li and RenYue Xiao(2007)] are taken as 50 and 300 respectively.

IV. Results And Discussions

The initial population is created randomly and the objective function is calculated. The new sequence generation based on the initial sequence illustrated in the following example. Consider the following initial sequence P_{ibest} and P_{gbest} as follows:

Present: 2 6 3 5 4 1
Pibest: 6 1 2 5 3 4
Pgbest: 5 3 6 4 2 1

Assume $C_1=C_2=1$ and rand () =0.57.Then Pibest is generated by swapping the individuals of a present sequence.

Present: 2 6 3 5 4 1 Swap :(2 6)

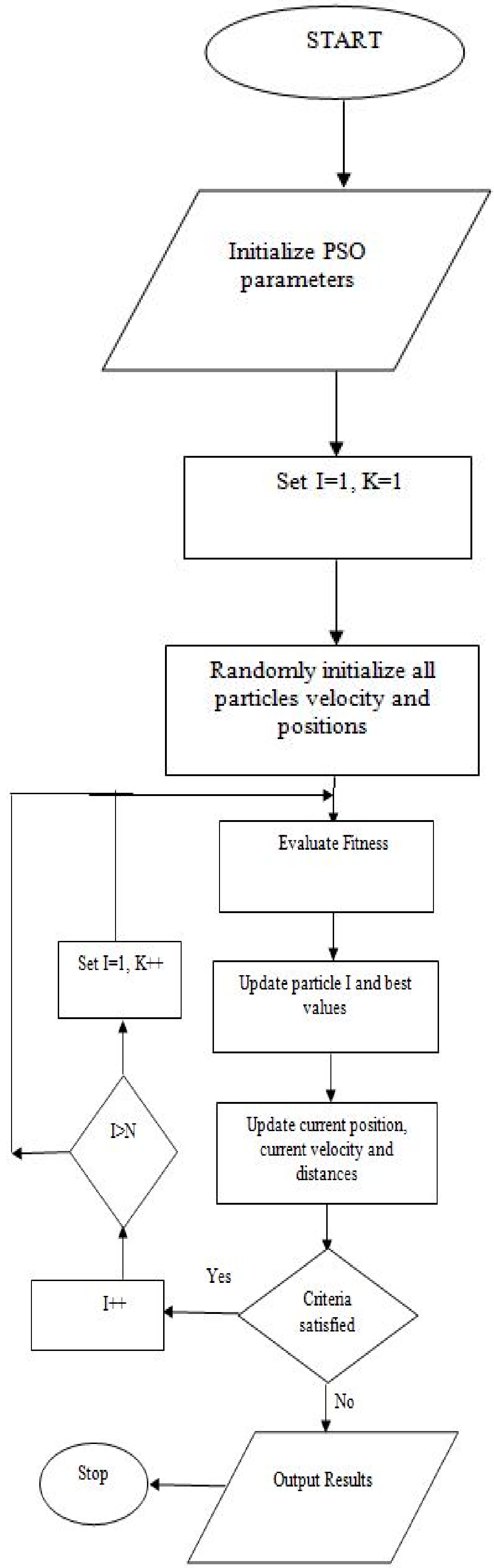


Fig 2: Flow chart of PSO algorithm

6 2 3 5 4 1 Swap :(2, 1)
6 1 3 5 4 2 Swap :(3, 2)

DOI: 03.LSCS.2013.1.502
ACEEE

6 1 2 5 4 3 Swap :(4, 3)

6 1 2 5 3 4 ——Pibest

Here (2,6) (2,1) (3,2) (4,3) are used for getting Pibest from the present sequence .The Pgbest is generated by swapping the individual of a present sequence.

Present: 2 6 3 5 4 1 Swap :(2, 5)

5 6 3 2 4 1 Swap: (6, 3)

5 3 6 2 4 1 Swap: (2, 4)

5 3 6 2 4 1—Pgbest

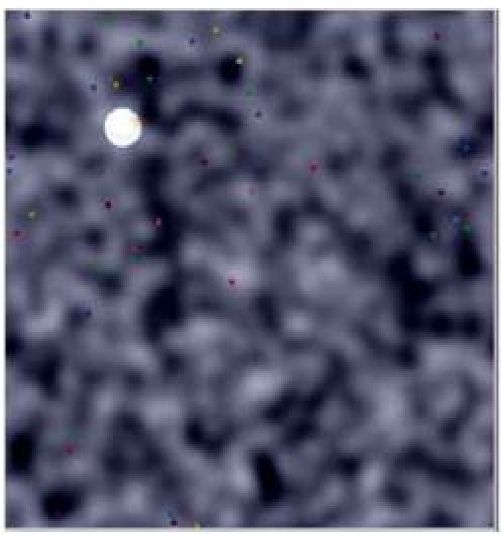
Hence (2, 5), (6, 3) and (2, 4) are used for getting Pgbest from present sequence.

Velocity=1*0.57{(2,6),(2,1),(3,2),(4,3)}+1*0.57{(2,5),(6,3),(2,4)}

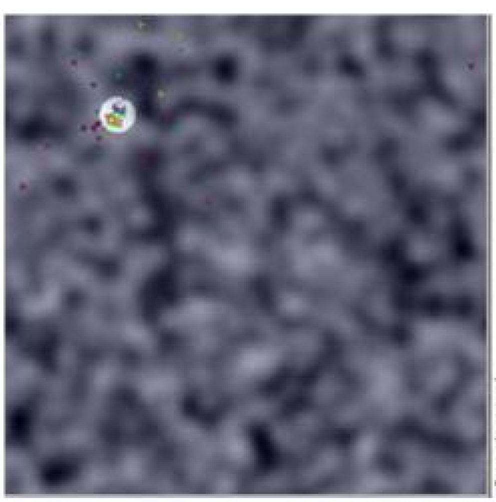
The 57% of the change in both the parts are considered. Hence the first two changes in both the parts (2,6),(2,1) and (2,5),(6,3) is considered.

New sequence=present+ velocity=2 6 3 5 4 1+ (2, 6), (2, 1), (2, 5), (6, 3)

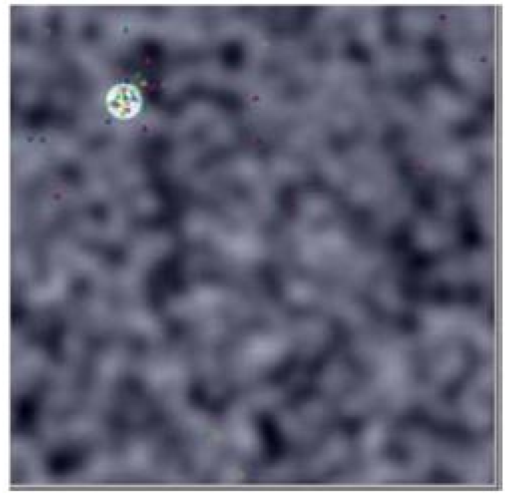
Hence the sequence is generated for the next generation is 3 1 6 2 4 5.Similarly for all other particles the new sequences are generated and objective function is evaluated and is shown in Fig.3 and Fig. 4.



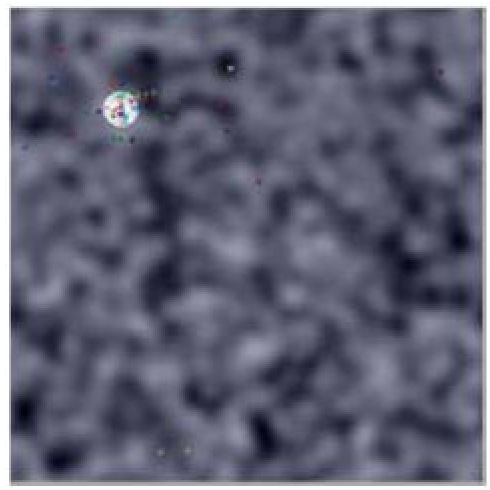
(a)



(b)



(c)



(d)

Fig 3: PSO's Particle location of the target (a) Randomly distributed particles (b) Particles position after 50 interactions (c) Particles position after 100 interactions

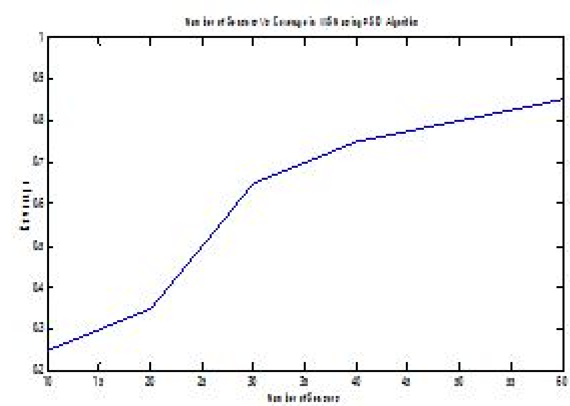


Fig 4: Sensor Vs Coverage

It is proved theoretically that the maximum efficient coverage [Men-Shen Tsai and Wu-chang Wu (2008)] or the minimum number of nodes required to cover rectangular area of 50×50 is 40.

DOI: 03.LSCS.2013.1.502

V. Conclusions

Scale and density of deployment, environmental uncertainties and constraints in energy, memory and bandwidth and computing resources pose serious challenge to the developer of WSNs. Most analytical methods suffer from slow or lack of convergence to the final solution. This call for first optimization algorithm that produces quality solution utilizing less resource.PSO has been popular technique used to solve optimization problem in WSN due to its simplicity, high quality of solution, first convergence and insignificant computational burden. The proposed work has the ability to achieve optimal solution of converge problem with minimum number of sensors in wireless network. The results show that PSO approach is effective and robust for efficient coverage problem of sensor deployment and is considered to give almost the optimal solutions in WSN. In future the focus can be given to achieve 100 % coverage with minimum number of sensors. The study of the 100% coverage using various optimal search techniques also presents several interesting challenges.

References

[1] J. Yick, B.Mukherjee, and D.Ghosal, *"Wireless Sensor Network Survey"*, computer networks, Vol 52,no-12,pp. 2292–2330,*2008*

[2] Eberhart R. and Kennedy J. (1995). *A new optimizer using Particle Swarm Theory.* Proc 6th Int. symposium on Micro machine and Human Science, pp.39-43

[3] Poli, R. 2007. Analysis of the publications on the Applications of Particle Swarm Optimisation, *Journal of Artificial Evolution and Applications*, **2008**, 1-10.

[4] Lazinica, A. 2009. *Particle Swarm Optimization.* (Edited), Published by In-Tech, Croatia, January

[5] K.V. Price, R.M Storn, and J.A Lampinen, *Differential evolution: A Practical approach to Global optimization, Ser. Natural Computing series*; Berlin, Germany: Springer-Verlag, 2005 .

DOI: 03.LSCS.2013.1.502